How Much Did It Rain? (L1 version)

Zhizhong Li
Nov 3, 2016
Information Engineering, CUHK
1155070507, lz015@ie.cuhk.edu.hk

1 Introduction

This report use sketching methods to deal with the least absolute deviation regression problem which minimizes

$$f(x) = \frac{1}{N}\|Ax - b\|_1, \tag{1}$$

where N is the number of samples, $A \in \mathbb{R}^{n\times d}$, $b \in \mathbb{R}^n$, and $x \in \mathbb{R}^d$. The two sketching methods are *Cauchy* and *Exponential*.

1.1 Dataset

Same as *HW1*, we use the *how much did it rain ii* [1, 2] dataset from *kaggle.com* as our benchmark. The aim of the dataset is to predict the amount of raining. The original dataset contains 13,765,201 training samples with 23 features and 1 prediction value. The following description was extracted from web page [1].

Id A unique number for the set of observations over an hour at a gauge.

Minutes_past For each set of radar observations, the minutes past the top of the hour that the radar observations were carried out. Radar observations are snapshots at that point in time.

Radardist_km Distance of gauge from the radar whose observations are being reported.

Ref Radar reflectivity in km.

Ref_5x5_10th, Ref_5x5_50th, Ref_5x5_90th 10th, 50-th, 90-th percentile of reflectivity values in 5x5 neighborhood around the gauge.

RefComposite, RefComposite_5x5_10th, RefComposite_5x5_50th, RefComposite_5x5_90th Maximum reflectivity in the vertical column above gauge. In dBZ.

RhoHV, RhoHV_5x5_10th, RhoHV_5x5_50th, RhoHV_5x5_90th Correlation coefficient.

Zdr, Zdr_5x5_10th, Zdr_5x5_50th, Zdr_5x5_90th Differential reflectivity in dB.

Kdp, Kdp_5x5_10th, Kdp_5x5_50th, Kdp_5x5_90th Specific differential phase (deg/km).

Expected Actual gauge observation in mm at the end of the hour.

Since it contains null values and outliers, we will do some preprocesses. All the experiments were conducted on a machine that has an 16-core Intel Xeon E5-2637 v2 CPU at 3.5GHz, and 256G memory. Codes were implemented using the Julia language and are available at *https://github.com/innerlee/random.computation.report1*.

1.2 Preprocess and Baselines

Since some observations are not complete, we first filter out 2,769,088 samples that do not contain missing values. The first two features *id* and *minutes_past* are used just for identification rows, thus

This is the *Randomness and Computation* HW2 report.

we omit them and get data matrices

$$A_0 \in \mathbb{R}^{2769088\times 21},\ b_0 \in \mathbb{R}^{2769088}. \tag{2}$$

The 99-th percentile of prediction values in b_0 is 144.0. We then use samples that have prediction values less than 144.0 and get new data

$$A_1 \in \mathbb{R}^{2741220\times 21},\ b_1 \in \mathbb{R}^{2741220}. \tag{3}$$

Since the LP solver for getting the exact solution cannot process large scale data in a reasonable time, we then truncate the matrix to 20,000 rows. As suggested by instruction, we normalize matrix A by subtracting mean and divide by std column-wisely. b is *not* normalized. The final dataset we use is

$$A \in \mathbb{R}^{20000\times 21},\ b \in \mathbb{R}^{20000}. \tag{4}$$

Solving problem (1) by LP solver, we have $x^* = (0.637049, -0.0401072, 0.117485, -0.196894, -0.375371, -0.267876, 0.855095, 0.509979, 0.146945, 0.0406474, -0.306016, 0.0689336, -0.0244424, 0.0619177, 0.202188, 0.395404, -0.183876, -0.0417801, 0.0202723, -0.00508027, -0.125157)$ with loss $f(x^*) = 3.429$ in 97.3 seconds. This serves as the baseline for our later discussion.

2 Sketching

Let $n = 20000$ and $d = 21$ as in our dataset. The general framework contains three steps:

1. L1-norm subspace embedding. This step can use *Cauchy* or *Exponential* mehtod. The output is a probability vector p. There are two parameters in this step. The first is the number of rows in *Cauchy* and *Exponential* subspace embedding. Some guidance on chosing this number are $\mathcal{O}(d\log(d))$ as in [3, Theorem 36] and $d \cdot \text{poly}(\log(d))$ as in [3, Theorem 41]. We fix this number to $21 \times \log(21) \approx 64$. The other is the columns of Gaussian sketching which is used for accelerating algorithm. We fix it to be 16.
2. Sampling. This step is similar like the Leverage score sketching. After this step, we get a shirinked sized problem. The number in sampling is controlled by parameter r. We tested $r = 100$ and $r = 500$ for reference.
3. Solving small sized problem. We use LP solver to get the solution.

Testing results are shown in Table 1.

Table 1: The performances of different Sketching techniques. r is the parameter in step 2. time-i is the average time spent on step i.

algorithm	repeat	r	time1	time2	time3	time	min loss	max loss	median loss	mean loss	std loss	rel err
Baseline	1	-	-	-	-	97.30	3.428	3.428	3.428	3.428	-	-
Cauchy	100	100	0.129	0.014	0.010	0.154	3.5507	4.7928	3.9607	3.9855	0.2484	3.56%
Cauchy	100	500	0.131	0.014	0.125	0.272	3.4497	3.5875	3.5023	3.5045	0.0259	0.62%
Exponential	100	100	0.012	0.002	0.011	0.025	3.6121	4.6369	3.8575	3.9185	0.2231	5.35%
Exponential	100	500	0.014	0.002	0.129	0.146	3.4486	3.5924	3.4934	3.4971	0.0251	0.58%

2.1 Cauchy

The minimum point when $r = 500$ is $x^* = (0.657501, 0.0776499, -0.133653, -0.0633284, -0.876821, -0.759774, 1.41801, 0.980646, 0.0688024, -0.0893941, -0.221192, 0.0627549, 0.0966104, 0.0311747, 0.359389, 0.235925, -0.00986104, 0.144246, -0.0530562, -0.0743144, -0.19942)$ with loss $f(x^*) = 3.4497$.

2.2 Exponential

The minimum point when $r = 500$ is $x^* = (0.493382, 0.42278, 0.242891, -0.476517, -0.753511, -0.43696, 0.680251, 0.761099, 0.196092, -0.137095, -0.35945, 0.181037, -0.033585, -0.037471, 0.506696, 0.282556, -0.221777, 0.00231164, 0.0114419, -0.144334, -0.0618985, -1.0)$ with loss $f(x^*) = 3.4486$.

3 Discussion

From the experiment results, we can see that

- Solving the problem using LP requires lots of time. and it is infeasible for large scale problems. In our experiments, we were forced to truncate the dataset to a small scale (20000 rows) due to this limitation.
- The preparation time for the sketching is negligible in experiments. This is different from HW1.
- The performances of the two sketching methods are similar. Increasing the number of rows sampled helps imporve performance. We can get a reletive error of 0.58% by sampling 2.5% of all rows.
- Time consumption of sketching methods is extremely small. This makes it suitable for large scale dataset.

Therefore sketching methods is very useful for large scale problems.

References

[1] Kaggle.com. How much did it rain? ii, oct 2016. URL `https://www.kaggle.com/c/how-much-did-it-rain-ii/`.

[2] V. Lakshmanan, A. Kleeman, J. Boshard, R. Minkowsky, and A. Pasch. The ams-ai 2015-2016 contest: Probabilistic estimate of hourly rainfall from radar. In *13th Conference on Artificial Intelligence*, Phoenix, AZ, 2015. American Meteorological Society.

[3] D. P. Woodruff. Sketching as a tool for numerical linear algebra. *arXiv preprint arXiv:1411.4357*, 2014.